

Máster Avanzado de Programación en Python para Hacking, BigData y Machine Learning

FUNDAMENTOS DE PYTHON

LECCIÓN 03

INSTALAR Y ADMINISTRAR IDEs. MI PRIMER PROGRAMA EN PYTHON.

ÍNDICE

Introducción

Objetivos

Principales IDEs para programar en Python

Instalación de Spyder en Ubuntu y Windows

Instalación de Visual Studio Code en Ubuntu y Windows

Principales librerías. Mi primer programa

Conclusiones

INTRODUCCIÓN

En esta lección mostraremos los principales entornos de desarrollo (IDEs) para desarrollar nuestro código en el lenguaje de programación en Python.

Adicionalmente, haremos un breve repaso sobre las principales librerías utilizadas en Python, haciendo uso de ellas en nuestro primer programa implementado en Python.

OBJETIVOS

Al finalizar esta lección serás capaz de:

- 1 Conocer los principales IDEs para programar en Python.
- 2 Saber instalar y configurar los IDEs en diferentes sistemas operativos.
- 3 Conocer las principales librerías de Python.
- 4 Saber ejecutar un programa en Python.

Principales IDEs para programar en Python

- Un entorno de desarrollo integrado o IDE es una aplicación informática que proporciona un conjunto de servicios para facilitar al programar el desarrollo de software.
- Podemos encontrar una gran variedad de IDEs que pueden facilitarnos la tarea de desarrollar código, y la elección del más adecuado es una tarea crucial, ya que nos facilitará:
 - Desarrollar código
 - Ejecución y depuración de errores.

Principales IDEs para programar en Python

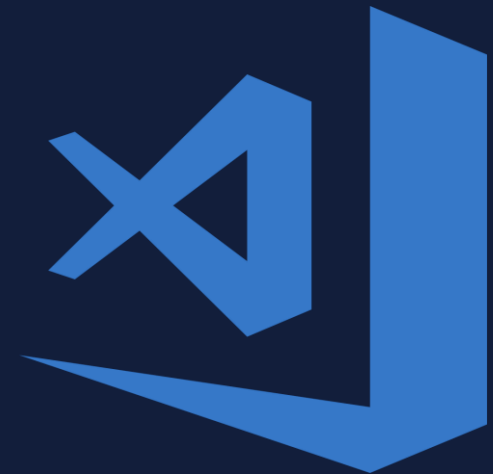
- Entorno de desarrollo integrado de código abierto y gratuito para programación científica en el lenguaje Python.
- Ofrece un nivel avanzado de edición, depuración y funciones de explorador de datos. Además, es compatible con la consola ipython.
- Fecha de lanzamiento: 18 de Octubre de 2009.
- Autor: Pierre Raybaut.



Principales IDEs para programar en Python

- Editor de código desarrollado por Microsoft para Windows, Linux y macOS.
- Entre sus principales ventajas se destaca el soporte para la **depuración**, control integrado de **Git** y **finalización inteligente de código** entre otros.
- Fecha de lanzamiento: Abril de 2015.
- Autor: Microsoft

Visual Studio Code



Principales IDEs para programar en Python

- Es uno de los IDE más completos y populares entre los desarrolladores de Python.
- Incluye **funciones inteligentes** que facilitan la tarea del programador, como un editor de código con sugerencias.
- Pycharm es un software de pago, aunque cuenta con una versión de prueba gratuita.
- Fecha de lanzamiento: Febrero de 2010.
- Autor: JetBrains.

Pycharm



Principales IDEs para programar en Python

- Entorno de desarrollo de código abierto.
- Incluye herramientas útiles, como el **autocompletado de código**, **sangrados inteligentes** y un **depurador**.
- Es un complemento de terceros de Eclipse
- Fecha de lanzamiento: Julio de 2003.
- Autor: Appcelerator.



Principales IDEs para programar en Python

- **IDLE**
 - entorno de desarrollo integrado para Python, **incluido** de forma **predeterminada** al instalar **Python** (desde la versión 1.5.2b1)
 - Incluye herramientas como el **resaltado de sintaxis**, **sangría inteligente** y un **depurador** con puntos de interrupción persistentes.
 - Autor: Guido van Rossum.

Instalación Spyder en Ubuntu 16.04

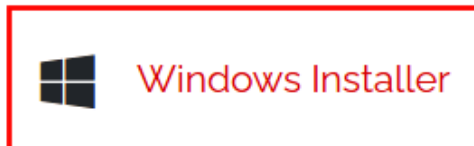
```
ramon@ramon-rd: ~  
(master_env) ramon@ramon-rd:~$ pip install spyder  
Collecting spyder  
  Downloading spyder-4.2.1-py3-none-any.whl (10.9 MB)  
    | ████████████████████████████████████████ | 10.9 MB 3.6 MB/s  
Collecting sphinx>=0.6.6  
  Downloading Sphinx-3.5.1-py3-none-any.whl (2.8 MB)  
    | ████████████████████████████████████████ | 2.8 MB 6.4 MB/s  
Collecting pyxdg>=0.26; platform_system == "Linux"  
  Downloading pyxdg-0.27-py2.py3-none-any.whl (49 kB)  
    | ████████████████████████████████████████ | 49 kB 4.2 MB/s  
Collecting parso==0.7.0  
  Downloading parso-0.7.0-py2.py3-none-any.whl (100 kB)  
    | ████████████████████████████████████████ | 100 kB 5.8 MB/s  
Collecting spyder-kernels<1.11.0,>=1.10.1  
  Downloading spyder_kernels-1.10.1-py2.py3-none-any.whl (64 kB)  
    | ████████████████████████████████████████ | 64 kB 3.0 MB/s  
Collecting pickleshare>=0.4  
  Downloading pickleshare-0.7.5-py2.py3-none-any.whl (6.9 kB)  
Collecting cloudpickle>=0.5.0  
  Downloading cloudpickle-1.6.0-py3-none-any.whl (23 kB)  
Collecting jedi==0.17.2  
  Downloading jedi-0.17.2-py2.py3-none-any.whl (1.4 MB)  
    | ████████████████████████████████████████ | 1.4 MB 7.1 MB/s  
Collecting pyls-black>=0.4.6
```

Instalación Spyder en Windows 10

- Accedemos a la web de Spyder para descargar el IDE:
<http://docs.spyder-ide.org/current/installation.html>.

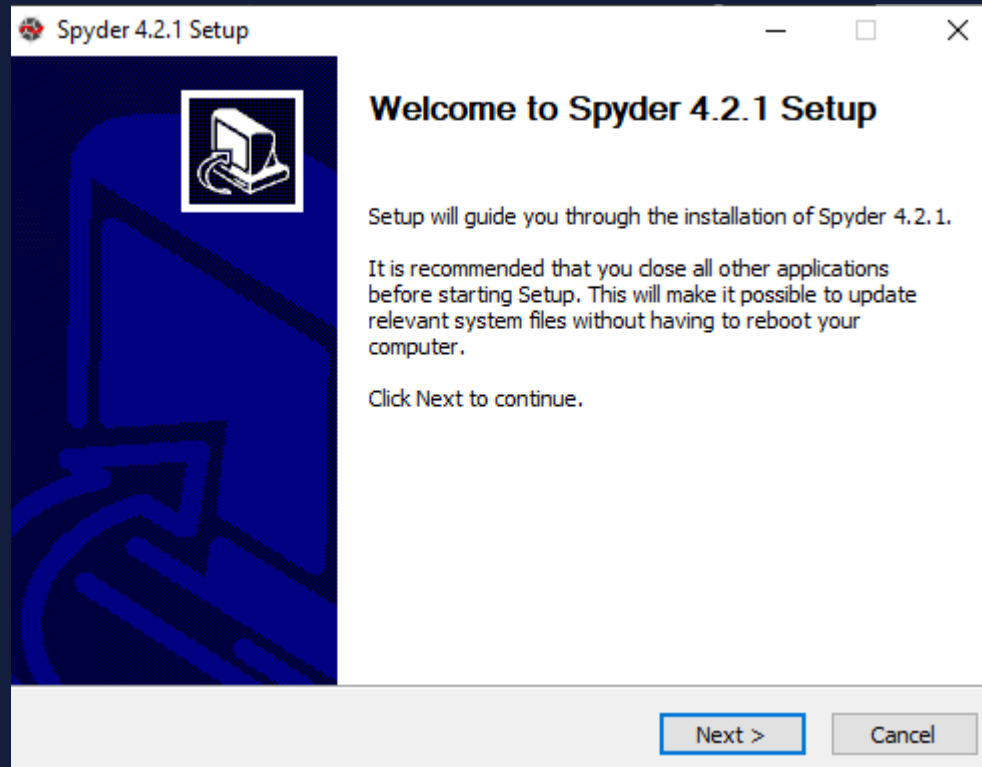
Standalone installers

Our standalone installers for Windows and macOS are available from Spyder 4.2 onwards. We recommend using this installation method on those platforms, but we offer several other options for Linux, advanced users and specific needs, so keep reading if that's the case for you.

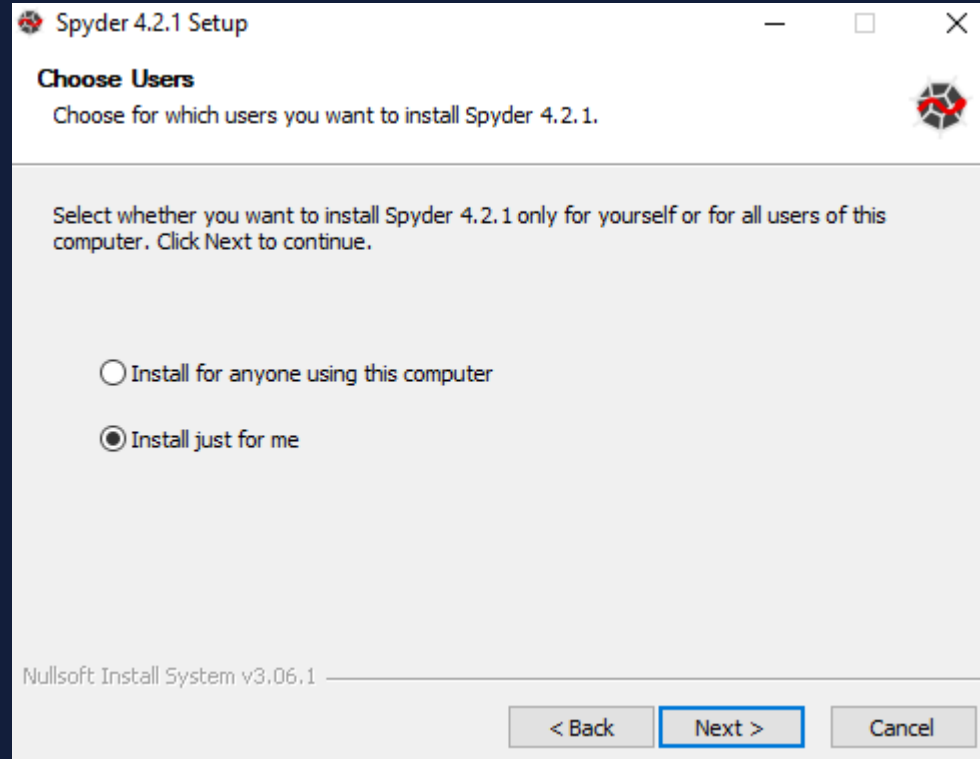


macOS Installer

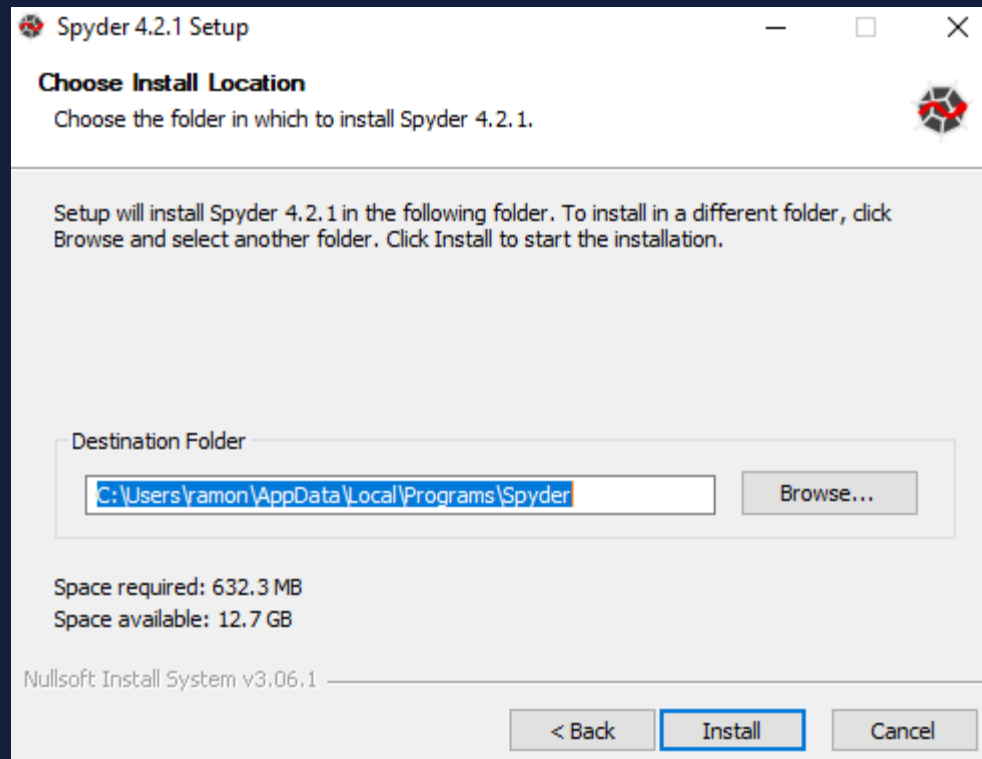
Instalación Spyder en Windows 10



Instalación Spyder en Windows 10

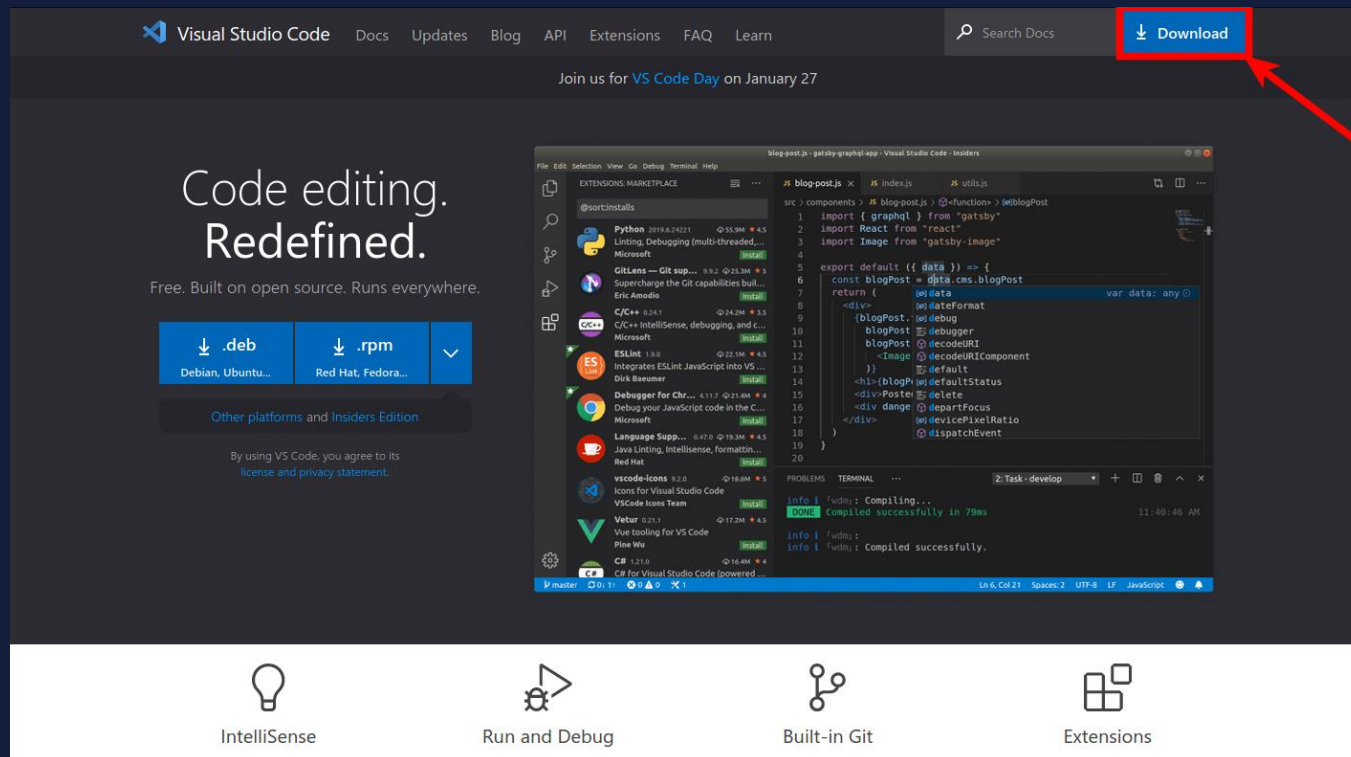


Instalación Spyder en Windows 10



Instalación Visual Studio Code en Ubuntu 16.04

En primer lugar, nos dirigimos a la web oficial de Visual Studio Core: <https://code.visualstudio.com/> y accedemos a la sección de descargas para obtener la versión adecuada.



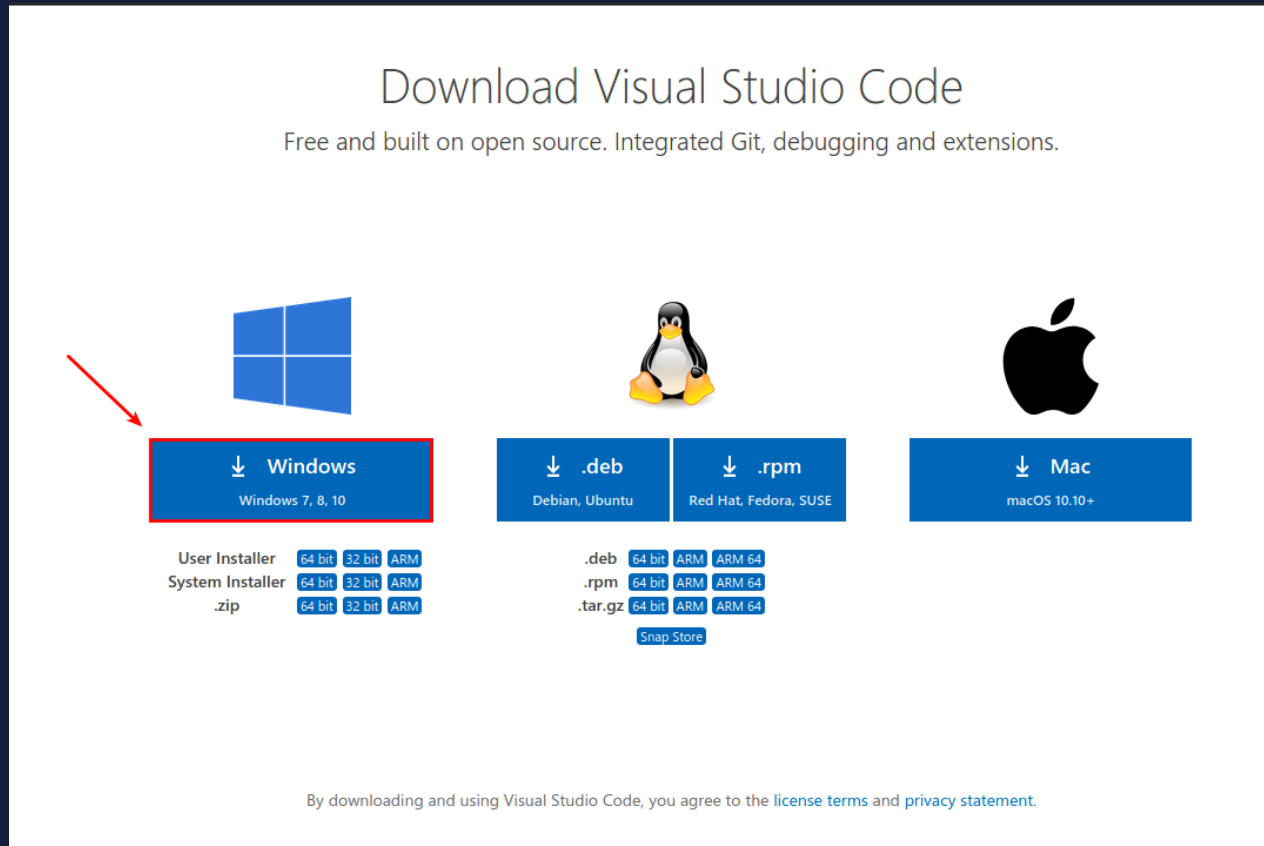
Instalación Visual Studio Code en Ubuntu 16.04

```
ramon@ramon-rd: ~/Downloads
ramon@ramon-rd:~/Downloads$ sudo dpkg -i code_1.52.1-1608136922_amd64.deb
Selecting previously unselected package code.
(Reading database ... 566221 files and directories currently installed.)
Preparing to unpack code_1.52.1-1608136922_amd64.deb ...
Unpacking code (1.52.1-1608136922) ...
Setting up code (1.52.1-1608136922) ...
Processing triggers for desktop-file-utils (0.22-1ubuntu5.2) ...
Processing triggers for bamfdaemon (0.5.3~bZR0+16.04.20180209-0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...
Processing triggers for gnome-menus (3.13.3-6ubuntu3.1) ...
Processing triggers for mime-support (3.59ubuntu1) ...
Processing triggers for shared-mime-info (1.5-2ubuntu0.2) ...
Unknown media type in type 'all/all'
Unknown media type in type 'all/allfiles'
ramon@ramon-rd:~/Downloads$
```

Instalación Visual Studio Code en Windows 10

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



The image shows the Visual Studio Code download page. It features three main sections for different operating systems: Windows, Linux, and Mac. The Windows section is highlighted with a red box and a red arrow. Below the Windows section, there are links for 'User Installer', 'System Installer', and '.zip', each with buttons for '64 bit', '32 bit', and 'ARM'. The Linux section has buttons for '.deb' and '.rpm' for 'Debian, Ubuntu' and 'Red Hat, Fedora, SUSE', with additional buttons for '64 bit', 'ARM', and 'ARM 64'. The Mac section has a button for 'Mac' for 'macOS 10.10+'. At the bottom, there is a link to the 'Snap Store'.

↓ Windows
Windows 7, 8, 10

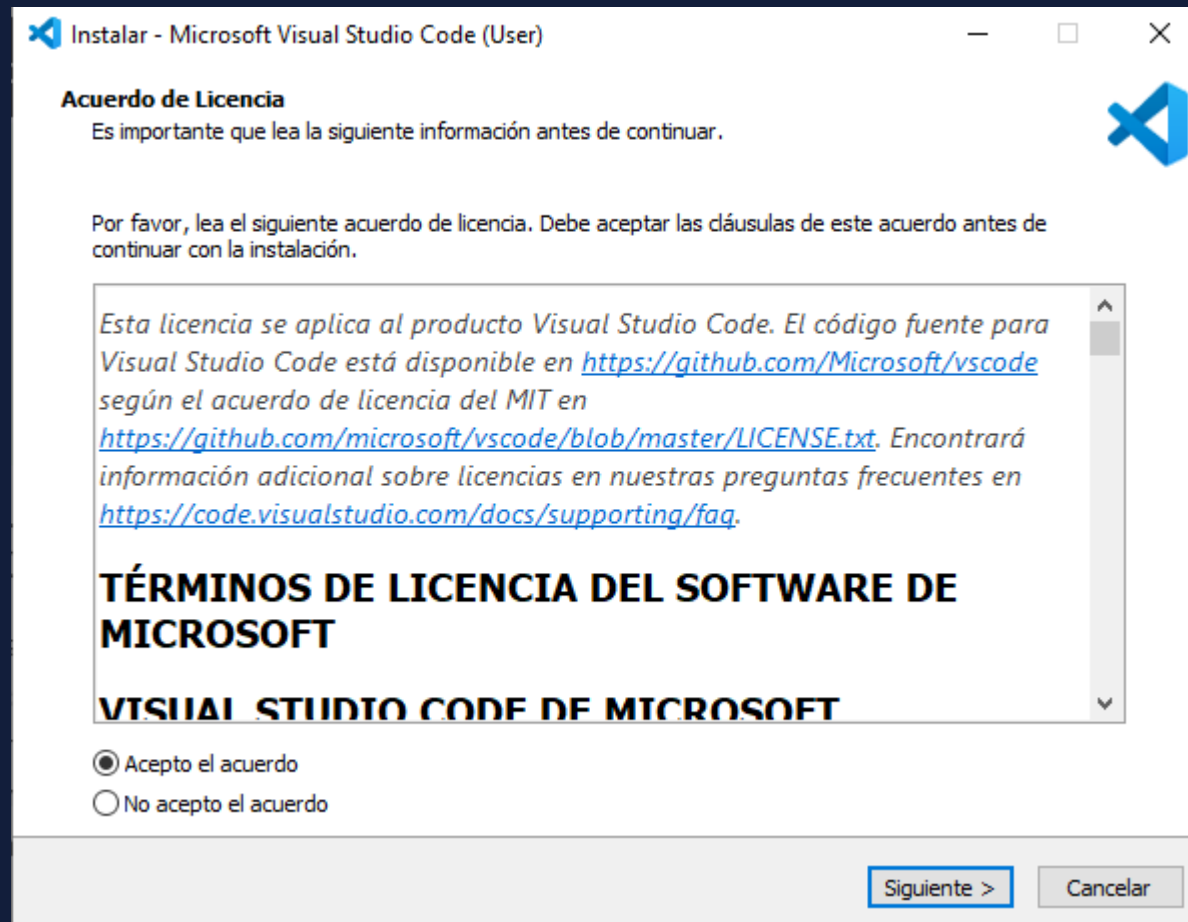
User Installer 64 bit 32 bit ARM
System Installer 64 bit 32 bit ARM
.zip 64 bit 32 bit ARM

↓ .deb Debian, Ubuntu 64 bit ARM ARM 64
↓ .rpm Red Hat, Fedora, SUSE 64 bit ARM ARM 64
.tar.gz 64 bit ARM ARM 64
Snap Store

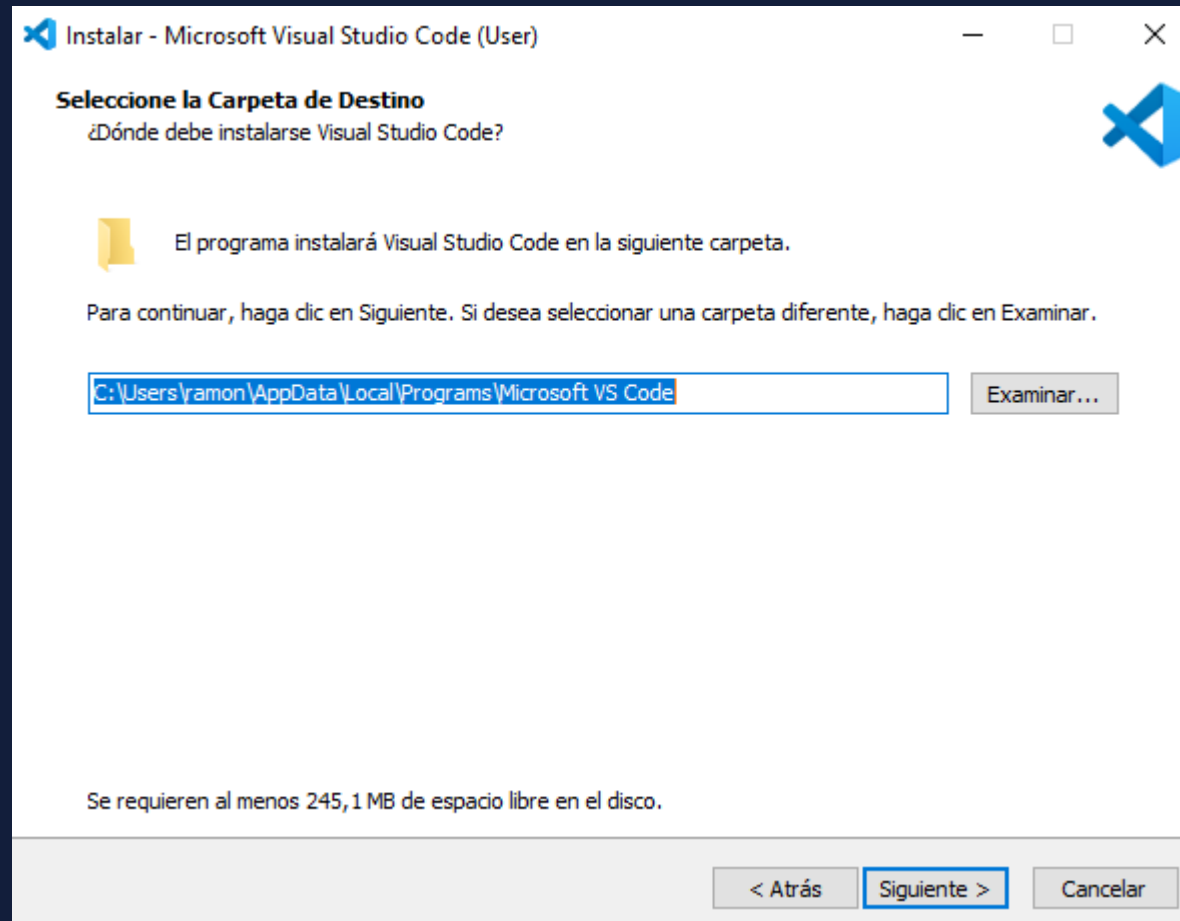
↓ Mac
macOS 10.10+

By downloading and using Visual Studio Code, you agree to the [license terms](#) and [privacy statement](#).

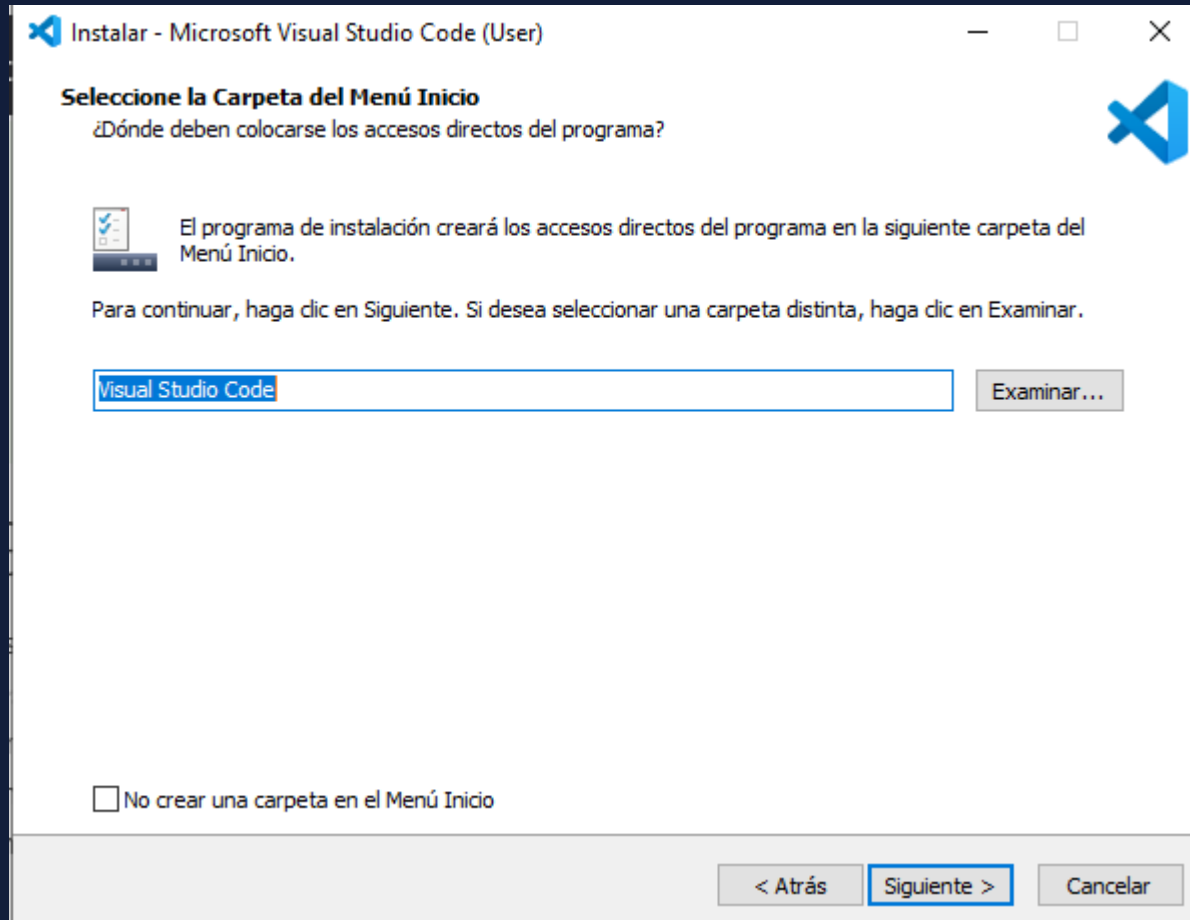
Instalación Visual Studio Code en Windows 10



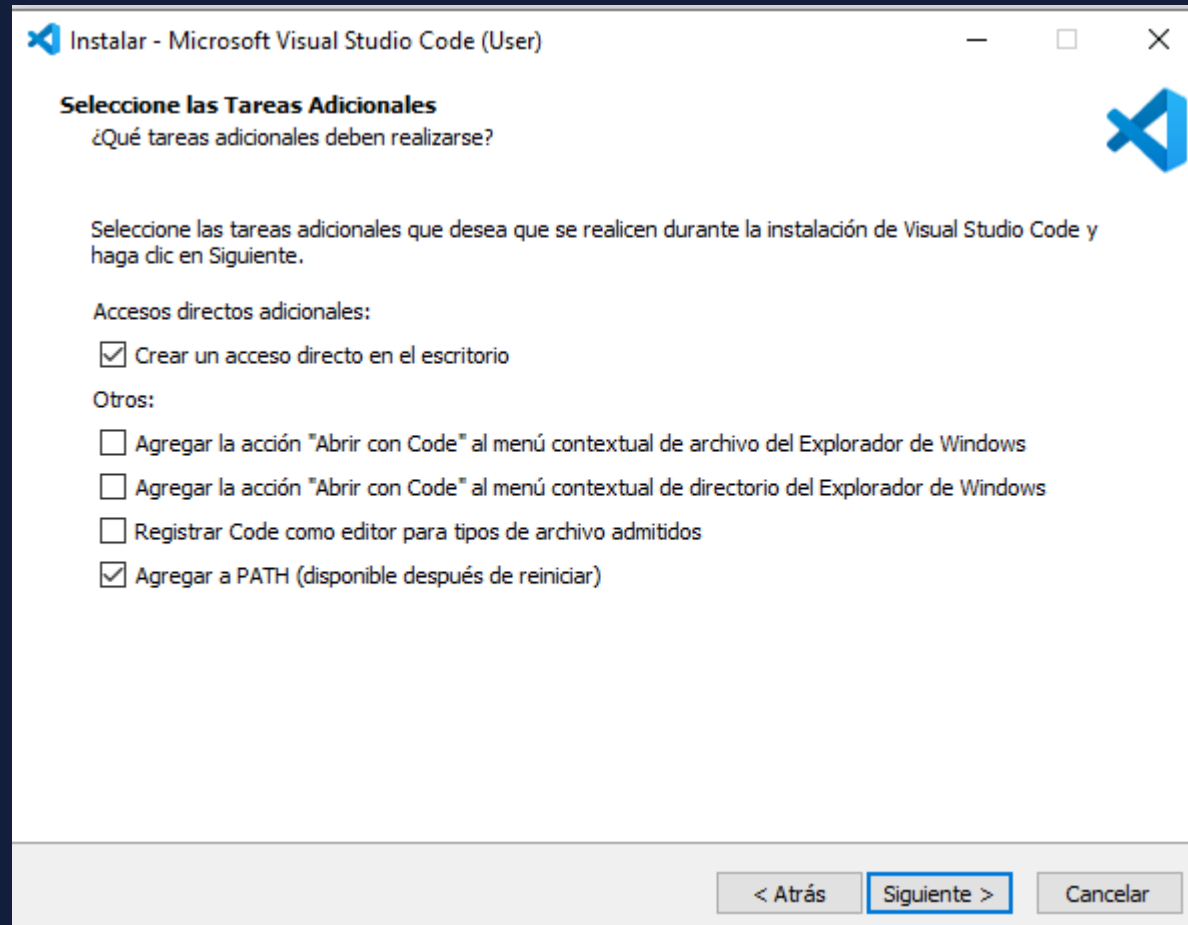
Instalación Visual Studio Code en Windows 10



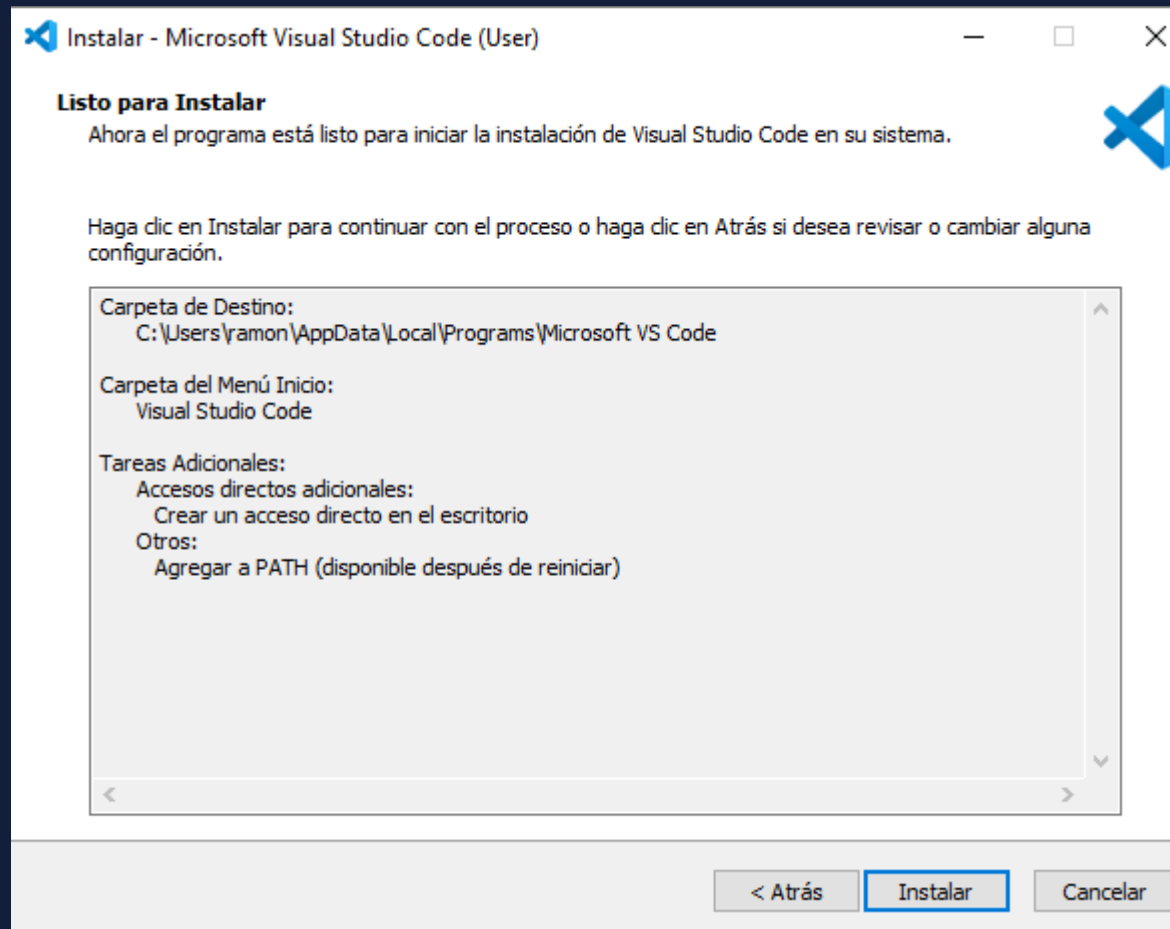
Instalación Visual Studio Code en Windows 10



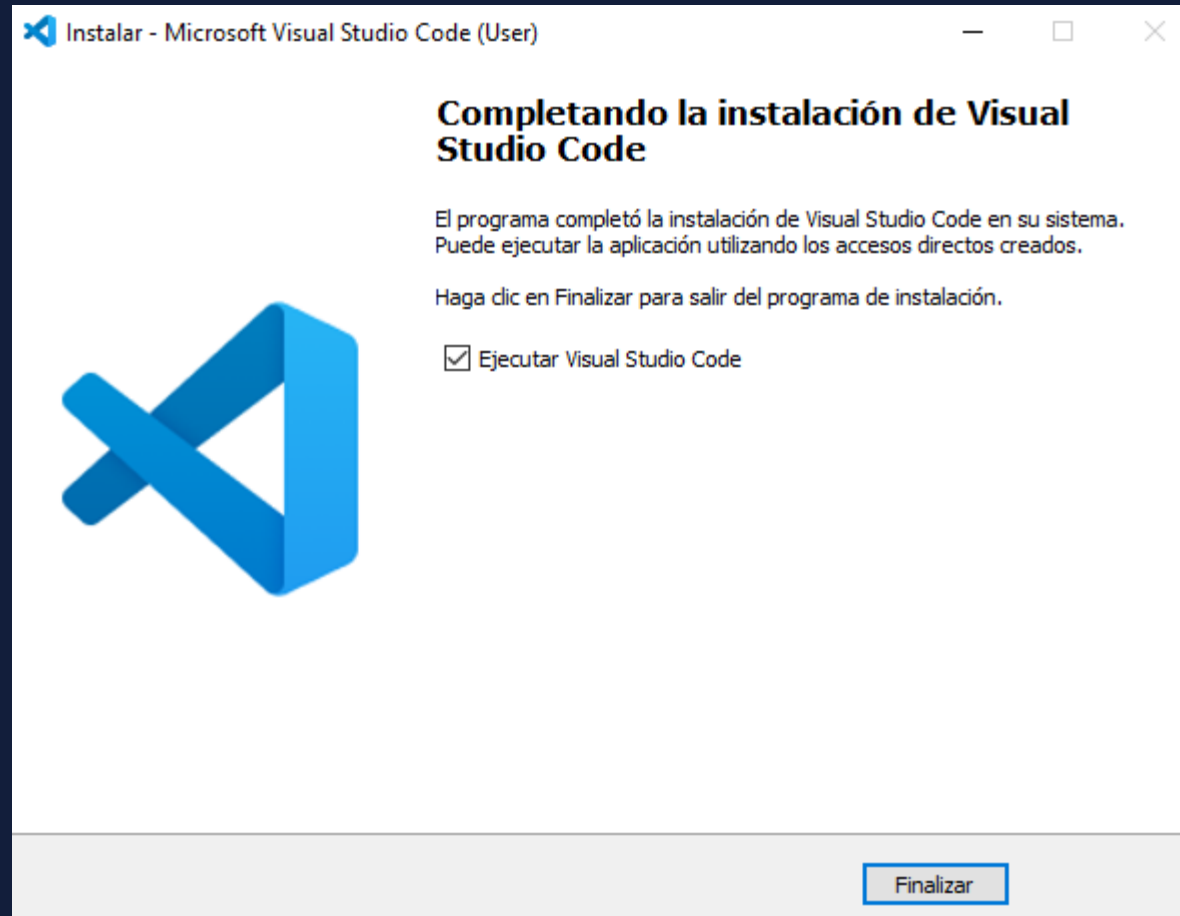
Instalación Visual Studio Code en Windows 10



Instalación Visual Studio Code en Windows 10



Instalación Visual Studio Code en Windows 10



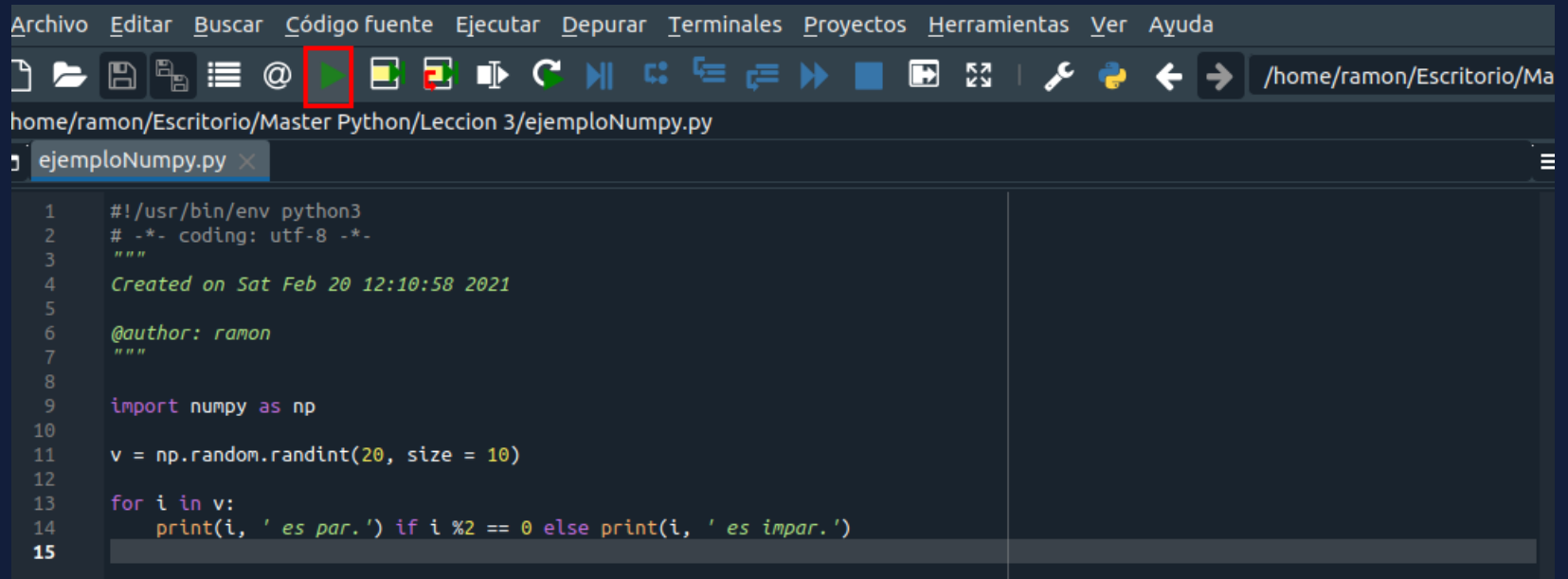
Principales librerías. Mi primer programa.

- Existen una gran multitud de librerías o paquetes disponibles en Python.
- Dependiendo del problema que el desarrollador quiera solucionar, podrá seleccionar la que más se adecúe a su tarea.
- Sin embargo, consideramos que hay dos librerías de Python principales que todo programador debe conocer. Hablamos de **numpy** y **matplotlib**.

Principales librerías. Mi primer programa.

- Numpy es el paquete fundamental para cálculo científico en Python.
- Es una biblioteca de Python que proporciona un objeto del tipo matriz multidimensional y una variedad de operaciones rápidas para realizar cálculos sobre matrices, incluyendo operaciones de entrada salida y álgebra lineal entre otros.
- La librería numpy no se instala por defecto al instalar una versión de Python, sino que tenemos que instalarla nosotros. (Ejecutando, por ejemplo, en un terminal: “pip install numpy”).

Principales librerías. Mi primer programa.



```
Archivo  Editar  Buscar  Código fuente  Ejecutar  Depurar  Terminales  Proyectos  Herramientas  Ver  Ayuda
/home/ramon/Escritorio/Master Python/Leccion 3/ejemploNumpy.py
ejemploNumpy.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Sat Feb 20 12:10:58 2021
5
6  @author: ramon
7  """
8
9  import numpy as np
10
11  v = np.random.randint(20, size = 10)
12
13  for i in v:
14      print(i, ' es par.') if i % 2 == 0 else print(i, ' es impar.')
15
```

Principales librerías. Mi primer programa.

The screenshot displays a code editor with a Python script and its execution results. The script, named `ejemploNumpy.py`, is located at `/home/ramon/Escritorio/Master Python/Leccion 3/ejemploNumpy.py`. It imports the `numpy` library and generates a random array of 10 integers. A loop then iterates over this array, printing each element and whether it is even or odd.

```
1  #!/usr/bin/env python3
2  #-*- coding: utf-8 -*-
3  """
4  Created on Sat Feb 20 12:10:50 2021
5
6  @author: ramon
7  """
8
9  import numpy as np
10
11  v = np.random.randint(20, size = 10)
12
13  for i in v:
14      print(i, ' es par.') if i % 2 == 0 else print(i, ' es impar.')
15
```

The execution output, shown in the terminal window, is as follows:

```
In [6]: runfile('/home/ramon/Escritorio/Master Python/Leccion 3/ejemploNumpy.py', wdir='/home/ramon/Escritorio/Master Python/Leccion 3')
16 es par.
15 es impar.
6 es par.
11 es impar.
9 es impar.
10 es par.
16 es par.
4 es par.
14 es par.
8 es par.

In [7]:
```

Nombr	Tipo	Tamaño	Valor
i	int64	1	8
v	Array of int64	(10,)	[16 15 6 11 9 10 16 4 14 8]

Principales librerías. Mi primer programa.

- Matplotlib es una librería desarrollada por John D. Hunter para la generación de gráficos a partir de datos contenidos en arrays en el lenguaje de programación Python y su extensión NumPy.
- La librería matplotlib no se instala por defecto al instalar una versión de Python, sino que tenemos que instalarla nosotros. (Ejecutando, por ejemplo, en un terminal: “pip install matplotlib”).

Principales librerías. Mi primer programa.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Feb 20 13:00:10 2021

@author: ramon
"""
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

labels = ['G1', 'G2', 'G3', 'G4', 'G5']
men_means = [20, 34, 30, 35, 27]
women_means = [25, 32, 34, 20, 25]

x = np.arange(len(labels)) # the label locations
width = 0.35 # the width of the bars

fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, men_means, width, label='Men')
rects2 = ax.bar(x + width/2, women_means, width, label='Women')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Scores')
ax.set_title('Scores by group and gender')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()

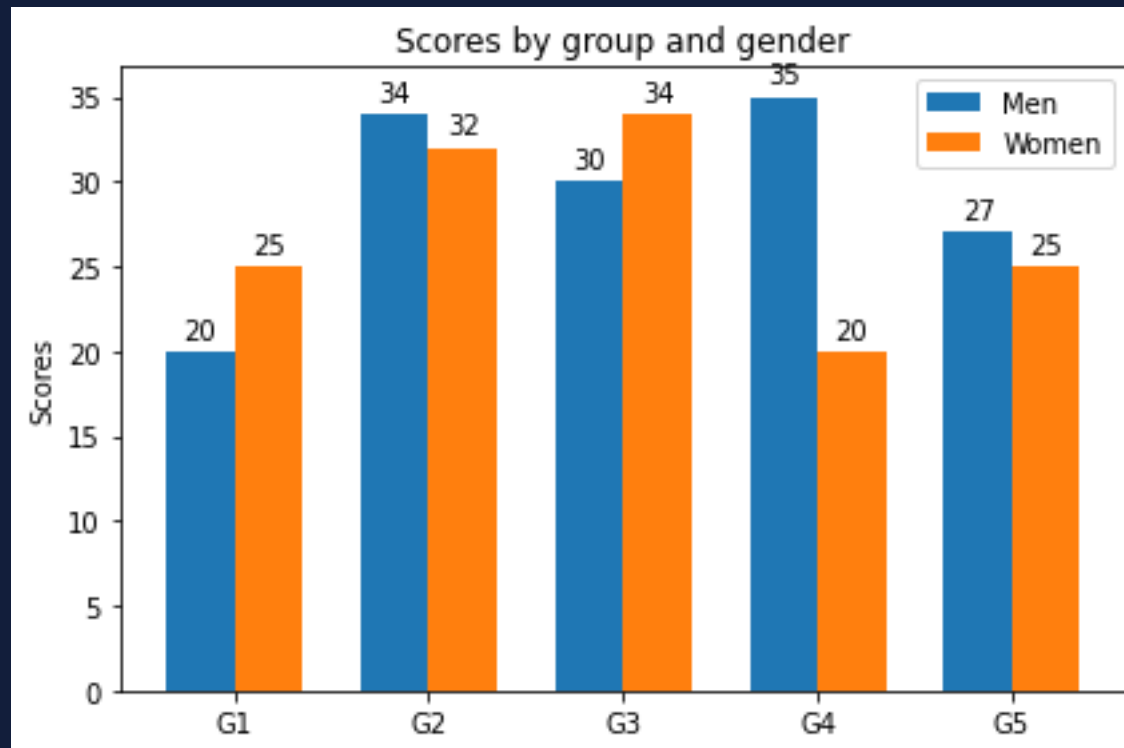
def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3), # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')

autolabel(rects1)
autolabel(rects2)

fig.tight_layout()

plt.show()
```

Principales librerías. Mi primer programa.





CONCLUSIONES

1

Debemos seleccionar el IDE que más se adapte a nuestras necesidades, pues facilitará nuestras tareas de programación.

2

Conocer las principales librerías de Python.

3

Implementar y ejecutar un programa utilizando los diferentes IDEs.

MUCHAS GRACIAS POR SU ATENCIÓN



ramonruedadelgado@gmail.com



Ramón Rueda Delgado
<https://www.linkedin.com/in/ramon-rueda/>



twitter.com/eiposgrados



facebook.com/eiposgrados



instagram.com/eiposgrados